Attention Guided Anomaly Localization in Images
Supplementary Material

Anonymous ECCV submission

Paper ID 2813

1 CAVGA is insensitive to anomaly localization threshold

Most baseline methods discussed in Sec. 4 (in the main paper) use anomalous training images to compute a threshold for localizing anomalies in images. However, as discussed in Sec. 3.1 (in the main paper), we empirically set 0.5 as our threshold for anomaly localization. To show that CAVGA-R_u is insensitive to the variations in threshold values we choose 0.2 through 0.6 as our threshold on attention map. We compare the localization performance of CAVGA-R_u for different threshold values to the best performing baseline method (γ-VAE_g) on the MvTAD dataset. We use the cell color in the quantitative result tables to denote the performance ranking in that row, where darker cell color means better performance. From Table 1, we observe that CAVGA-R_u is insensitive to the variations in threshold and that CAVGA-R_u outperforms γ-VAE_g in at least 9 out of 15 categories.

Table 1: Comparison of anomaly localization in IoU of CAVGA-R_u for different threshold values with the best baseline method (γ-VAE_g) on the MVTAD dataset

Method Threshold	γ-VAE_g	CAVGA-R_u 0.2	CAVGA-R_u 0.3	CAVGA-R_u 0.4	CAVGA-R_u 0.5	CAVGA-R_u 0.6
Bottle	0.27	0.29	0.28	0.29	0.34	0.31
Hazelnut	0.63	0.41	0.45	0.46	0.51	0.46
Capsule	0.24	0.25	0.24	0.27	0.31	0.28
Metal Nut	0.22	0.38	0.39	0.44	0.45	0.36
Leather	0.41	0.78	0.77	0.78	0.79	0.77
Pill	0.48	0.34	0.31	0.35	0.40	0.33
Wood	0.45	0.52	0.54	0.57	0.59	0.56
Carpet	0.79	0.70	0.69	0.71	0.73	0.71
Tile	0.38	0.28	0.26	0.31	0.38	0.38
Grid	0.36	0.37	0.38	0.34	0.38	0.37
Cable	0.26	0.38	0.41	0.41	0.44	0.38
Transistor	0.44	0.27	0.25	0.31	0.35	0.29
Toothbrush	0.37	0.52	0.51	0.53	0.57	0.53
Screw	0.38	0.38	0.40	0.45	0.48	0.46
Zipper	0.17	0.23	0.25	0.26	0.26	0.25
mean	0.39	0.41	0.41	0.43	0.47	0.43

2 Implementation details

All the images of the MVTAD, mSTC and LAG datasets are randomly center cropped to 256×256 and randomly rotated between $[-15°, +15°]$ to create variations in data during training. We train CAVGA$_u$ and CAVGA$_w$ with a learning rate of $1e^{-4}$ with a batch size of 16 for 150 epochs. To stabilize the training, the learning rate is decayed by 0.1 for every 30 epochs. For the MNIST, CIFAR-10 and Fashion-MNIST datasets, we use the images of size 32×32 and follow the same data augmentation and training procedure as mentioned in Sec. 4 (in main paper).

3 Additional quantitative results

We use the cell color in the quantitative result tables to denote the performance ranking in that row, where darker cell color means better performance.

AuROC of β-VAE on the MNIST and CIFAR-10 datasets: We compare CAVGA-D$_u$ with the baseline methods in Table 7 (in main paper) along with β-VAE [6] as an additional baseline in terms of AuROC on MNIST and CIFAR-10 datasets for anomaly detection. From Table 2, we see that CAVGA-D$_u$ outperforms β-VAE on all classes on MNIST and 9 out of 10 classes on CIFAR-10 datasets. Since [5] does not provide the class wise performance of MemAE on MNIST and CIFAR-10 and only provide the mean AuROC, we compared CAVGA-D$_u$ with MemAE in terms of mean AuROC and reported the result in Sec. 5 (in main paper under "performance on anomaly detection").

IoU for all baseline methods on MVTAD dataset: We compare CAVGA$_u$ and CAVGA$_w$ with the baseline methods in Table 3 (in main paper) along with CNNFD [11], TI [3] and VM [14] as additional baselines in terms of IoU on each category of the MVTAD dataset for anomaly localization. From Table 3, we see that CAVGA-D$_u$ and CAVGA-R$_u$ outperform all baselines including CNNFD, TI and VM in terms of IoU and mean IoU.

Category specific AuROC on the MVTAD dataset: We compare CAVGA$_u$ and CAVGA$_w$ with the baseline methods in terms of AuROC on each category of the MVTAD dataset. Table 4 shows that CAVGA-D$_u$ and CAVGA-R$_u$ achieve comparable performance with the baseline methods AE$_{\text{SSIM}}$, γ-VAE$_{\text{g}}$ and ADVAE. Furthermore, CAVGA-D$_u$ and CAVGA-R$_u$ outperform the best performing baseline (AE$_{\text{SSIM}}$) in 11 out of 15 and 13 out of 15 categories with an improvement ranging from 1% to 6% and 1% to 21% respectively.

Table 2: Performance comparison of CAVGA-D$_u$ and β-VAE in terms of AuROC and mean AuROC on MNIST and CIFAR-10 datasets for anomaly detection

Dataset	Method	0	1	2	3	4	5	6	7	8	9	mean
MNIST [8]	β-VAE [6]	0.890	0.841	0.967	0.947	0.968	0.966	0.907	0.899	0.946	0.794	0.913
	CAVGA-D$_u$	**0.994**	**0.997**	**0.989**	**0.983**	**0.977**	**0.968**	**0.988**	**0.986**	**0.988**	**0.991**	**0.986**
CIFAR-10 [7]	β-VAE [6]	0.368	0.746	0.397	0.604	0.387	**0.611**	0.500	0.614	0.399	0.698	0.532
	CAVGA-D$_u$	**0.653**	**0.784**	**0.761**	**0.747**	**0.775**	0.552	**0.813**	**0.745**	**0.801**	**0.730**	**0.736**

Table 3: Performance comparison of anomaly localization in category specific IoU and mean IoU ($\overline{\text{IoU}}$) on the MVTAD dataset. The darker cell color indicates better performance ranking in each row

Category	AVID [12]	AE_{SSIM} [2]	AE_{L2} [2]	AnoGAN [13]	γ-VAE_g [4]	LSA [1]	ADVAE [10]	CNNDF [11]	TI [3]	VM [14]	CAVGA $-D_u$	CAVGA $-R_u$	CAVGA $-D_w$	CAVGA $-R_w$
Bottle	0.28	0.15	0.22	0.05	0.27	0.27	0.27	0.07	-	0.03	0.30	0.34	0.36	0.39
Hazelnut	0.54	0.00	0.41	0.02	0.63	0.41	0.44	0.00	-	-	0.44	0.51	0.58	0.79
Capsule	0.21	0.09	0.11	0.04	0.24	0.22	0.11	0.00	-	0.01	0.25	0.31	0.38	0.41
Metal Nut	0.05	0.01	0.26	0.00	0.22	0.38	0.49	0.13	-	0.19	0.39	0.45	0.46	0.46
Leather	0.32	0.34	0.67	0.34	0.41	0.77	0.24	0.74	0.98	-	0.76	0.79	0.80	0.84
Pill	0.11	0.07	0.25	0.17	0.48	0.18	0.18	0.00	-	0.13	0.34	0.40	0.44	0.53
Wood	0.14	0.36	0.29	0.14	0.45	0.41	0.14	0.47	0.51	-	0.56	0.59	0.61	0.66
Carpet	0.25	0.69	0.38	0.34	0.79	0.76	0.10	0.20	0.29	-	0.71	0.73	0.70	0.81
Tile	0.09	0.04	0.23	0.08	0.38	0.32	0.23	0.14	0.11	-	0.31	0.38	0.68	0.81
Grid	0.51	0.88	0.83	0.04	0.36	0.20	0.02	0.02	0.01	-	0.32	0.38	0.42	0.55
Cable	0.27	0.01	0.05	0.01	0.26	0.36	0.18	0.13	-	-	0.37	0.44	0.49	0.51
Transistor	0.18	0.01	0.22	0.08	0.44	0.21	0.30	0.03	-	-	0.30	0.35	0.38	0.45
Toothbrush	0.43	0.08	0.51	0.07	0.37	0.48	0.14	0.00	-	0.24	0.54	0.57	0.60	0.63
Screw	0.22	0.03	0.34	0.01	0.38	0.38	0.17	0.00	-	0.12	0.42	0.48	0.51	0.66
Zipper	0.25	0.10	0.13	0.01	0.17	0.14	0.06	0.00	-	-	0.20	0.26	0.29	0.31
$\overline{\text{IoU}}$	0.26	0.22	0.33	0.09	0.39	0.37	0.20	0.13	0.38	0.12	0.41	0.47	0.51	0.59

Table 4: Comparison of anomaly localization in AuROC and mean AuROC of CAVGA$_u$ and CAVGA$_w$ with state-of-the-art approaches on the MVTAD dataset

Category	AVID [12]	LSA [1]	ADVAE [10]	γ-VAE$_g$ [2]	AE$_{SSIM}$ [2]	AE$_{L2}$ [13]	AnoGAN [4]	CNNDF [11]	TI [3]	VM [14]	CAVGA -D$_u$	CAVGA -R$_u$	CAVGA -D$_w$	CAVGA -R$_w$
Bottle	0.79	0.81	0.87	0.83	0.93	0.86	0.86	0.78	-	0.82	0.85	0.89	0.92	0.91
Hazelnut	0.83	0.77	0.98	0.95	0.94	0.95	0.87	0.72	-	-	0.96	0.96	0.98	0.98
Capsule	0.88	0.63	0.74	0.93	0.93	0.88	0.84	0.84	-	0.93	0.94	0.95	0.96	0.96
Metal Nut	0.91	0.78	0.94	0.93	0.83	0.86	0.76	0.82	-	0.83	0.84	0.85	0.91	0.92
Leather	0.73	0.84	0.95	0.78	0.78	0.75	0.64	0.87	0.97	-	0.79	0.94	0.97	0.97
Pill	0.79	0.77	0.83	0.91	0.91	0.85	0.87	0.68	-	0.83	0.96	0.94	0.93	0.93
Wood	0.87	0.80	0.77	0.72	0.73	0.73	0.73	0.91	0.78	-	0.74	0.86	0.92	0.93
Carpet	0.85	0.74	0.78	0.83	0.85	0.59	0.54	0.72	0.88	-	0.88	0.88	0.90	0.91
Tile	0.83	0.82	0.80	0.79	0.79	0.51	0.50	0.93	0.41	-	0.61	0.80	0.95	0.95
Grid	0.77	0.79	0.73	0.85	0.94	0.90	0.58	0.59	0.72	-	0.82	0.95	0.95	0.94
Cable	0.74	0.81	0.90	0.87	0.82	0.86	0.78	0.79	-	-	0.84	0.85	0.92	0.91
Transistor	0.78	0.83	0.93	0.89	0.83	0.86	0.80	0.66	-	-	0.84	0.85	0.91	0.93
Toothbrush	0.80	0.88	0.94	0.87	0.88	0.93	0.90	0.77	-	0.68	0.89	0.91	0.94	0.95
Screw	0.68	0.83	0.97	0.82	0.96	0.96	0.80	0.87	-	0.94	0.83	0.85	0.96	0.97
Zipper	0.50	0.71	0.78	0.91	0.88	0.77	0.78	0.76	-	-	0.93	0.94	0.91	0.95
mean AuROC	0.78	0.79	0.86	0.86	0.87	0.82	0.75	0.78	0.75	0.84	0.85	0.89	0.94	0.94

Scene specific AuROC on the mSTC dataset: We compare CAVGA$_u$ and CAVGA$_w$ with the baseline methods in terms of AuROC on each scene of the mSTC dataset. From Table 5, we observe that CAVGA-D$_u$ and CAVGA-R$_u$ outperform the best performing baseline method (γ-VAE$_g$) in 8 out of 12 scenes with an improvement ranging from 1% to 30%. Furthermore, CAVGA-D$_w$ and CAVGA-R$_w$ also outperform the baseline methods in all scenes of the mSTC dataset.

Table 5: Comparison of AuROC for anomaly localization on the mSTC dataset for each scene ID and their mean

s_i	γ-VAE$_g$ [4]	AVID [12]	LSA [1]	AE$_{SSIM}$ [2]	AE$_{L2}$ [2]	CAVGA -D$_u$	CAVGA -R$_u$	CAVGA -D$_w$	CAVGA -R$_w$
01	0.73	0.65	0.77	0.63	0.72	0.69	0.71	0.78	0.81
02	0.82	0.68	0.89	0.71	0.64	0.75	0.80	0.90	0.89
03	0.81	0.79	0.83	0.83	0.64	0.84	0.85	0.87	0.88
04	0.76	0.63	0.69	0.94	0.91	0.93	0.94	0.95	0.95
05	0.88	0.77	0.75	0.90	0.83	0.89	0.91	0.92	0.92
06	0.90	0.71	0.85	0.87	0.80	0.91	0.92	0.91	0.92
07	0.87	0.83	0.86	0.88	0.89	0.88	0.92	0.93	0.93
08	0.89	0.89	0.73	0.61	0.68	0.75	0.83	0.84	0.86
09	0.88	0.90	0.88	0.53	0.75	0.89	0.89	0.91	0.90
10	0.73	0.63	0.67	0.41	0.31	0.60	0.65	0.74	0.85
11	0.69	0.85	0.90	0.88	0.85	0.90	0.90	0.93	0.93
12	0.85	0.87	0.88	0.93	0.89	0.91	0.93	0.94	0.95
mean	0.82	0.77	0.81	0.76	0.74	0.83	0.85	0.89	0.90

AuROC on the Fashion-MNIST dataset: Table 6 shows that CAVGA-D$_u$ outperforms the most competitive baseline (γ-VAE$_g$) in AuROC in the unsupervised setting on the Fashion-MNIST dataset [15]. Specifically, CAVGA-D$_u$ outperforms γ-VAE$_g$ in 8 out of 10 classes of the Fashion-MNIST dataset with an improvement ranging from 2% to 24%.

Table 6: Comparison of CAVGA-D$_u$ for anomaly detection in terms of AuROC and its mean with the state-of-the-art methods on the Fashion-MNIST dataset

Class	LSA [1]	γ-VAE$_g$ [4]	CapsNet$_{PP}$ [9]	CapsNet$_{RE}$ [9]	β-VAE [6]	CAVGA-D$_u$
0	0.755	0.750	0.620	0.454	0.500	0.768
1	0.841	0.866	0.851	0.871	0.860	0.884
2	0.883	0.810	0.818	0.486	0.459	0.861
3	0.906	0.933	0.895	0.693	0.730	0.947
4	0.854	0.931	0.790	0.394	0.379	0.755
5	0.983	0.934	0.691	0.982	0.985	0.952
6	0.766	0.655	0.801	0.480	0.501	0.814
7	0.844	0.930	0.619	0.787	0.842	0.945
8	0.980	0.971	0.912	0.885	0.876	0.991
9	0.918	0.977	0.656	0.754	0.701	0.930
mean	0.873	0.876	0.765	0.679	0.683	0.885

Complexity analysis: We compare CAVGA$_u$ with the baseline methods in terms of training and testing time (in hours) for anomaly localization and detection. From Table 7, we see that CAVGA is practically reasonable to train and test on a single NVIDIA GeForce GTX 1080Ti GPU having comparable training and testing time with the baseline methods.

Table 7: Training/testing time (in hours) needed on the entire training/testing set of MVTAD for anomaly localization and detection

method	AnoGAN	AVID	AE$_{\text{SSIM}}$	AE$_{\text{L2}}$	LSA	OCGAN	ULSLM	CAVGA-D	CAVGA-R
training	2.5	3.0	2.0	2.0	4.0	1.5	2.0	3.0	7.0
testing	0.2	0.3	0.1	0.1	0.3	0.1	0.1	0.4	0.6

Ablation study: As mentioned in Sec. 6 (in main paper), we perform the ablation studies on 15 categories of the MVTAD dataset. Table 8 shows the ablation for all 15 categories, which illustrates the effectiveness of the convolutional z in CAVGA, L_{ae} in the unsupervised setting, and L_{cga} in the weakly supervised setting.

Table 8: The ablation study showing anomaly localization in IoU on all 15 categories of the MVTAD dataset. CAVGA-R$^*_u$ and CAVGA-R$^*_w$ are our base architecture with a flattened z in the unsupervised and weakly supervised settings respectively. "conv z" means using convolutional z

Method / Category	CAVGA -R$^*_u$	CAVGA -R$^*_u$ + L_{ae}	CAVGA -R$_u$ + conv z	CAVGA -R$_u$ + conv z + L_{ae}	CAVGA -R$^*_w$	CAVGA -R$^*_w$ + L_{cga}	CAVGA -R$_w$ + conv z	CAVGA -R$_w$ + conv z + L_{cga}
Column ID	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
Bottle	0.24	0.27	0.26	0.33	0.16	0.34	0.28	0.39
Hazelnut	0.16	0.26	0.31	0.47	0.51	0.76	0.67	0.79
Capsule	0.09	0.22	0.14	0.31	0.18	0.36	0.27	0.41
Metal Nut	0.28	0.38	0.34	0.45	0.25	0.38	0.28	0.46
Leather	0.55	0.71	0.64	0.79	0.72	0.79	0.75	0.84
Pill	0.24	0.35	0.29	0.40	0.24	0.44	0.43	0.53
Wood	0.25	0.43	0.36	0.59	0.51	0.62	0.61	0.66
Carpet	0.48	0.59	0.53	0.73	0.69	0.78	0.72	0.81
Tile	0.07	0.18	0.23	0.32	0.66	0.77	0.73	0.81
Grid	0.15	0.27	0.24	0.32	0.31	0.48	0.51	0.55
Cable	0.30	0.38	0.36	0.43	0.47	0.58	0.51	0.63
Transistor	0.17	0.29	0.26	0.34	0.33	0.41	0.39	0.45
Toothbrush	0.41	0.46	0.49	0.55	0.54	0.61	0.60	0.66
Screw	0.11	0.18	0.34	0.48	0.16	0.24	0.22	0.31
Zipper	0.07	0.18	0.21	0.25	0.19	0.24	0.29	0.31
mean	0.24	0.34	0.33	0.47	0.39	0.52	0.48	0.60

4 Architectural details

Table 9: Architectural details of CAVGA-R_u and CAVGA-R_w. The notation in each row is as follows: operation, filter h × filter w, number of filters, stride, pad. W.S. denotes the additional layers for the weakly supervised setting. ConvTr 2D denotes transpose convolution layer, Conv 2D denotes convolution layer

network	layer name	layer dimensions	output dimensions
Encoder	Layer 1 - 18	pretrained Resnet-18 (convolution only)	8 × 8 × 512
	Layer 19	ReLU	8 × 8 × 512
	Layer 20	Conv 2D, 1 × 1, 512, 1,0	8 × 8 × 512
	Layer 21	Conv 2D, 1 × 1, 512, 1,0	8 × 8 × 512
	W.S.: Layer 22	Flatten	32768
	W.S.: Layer 23	Linear	2
	W.S.: Layer 24	Softmax	2
Decoder	Layer 1	ConvTr 2D, 4 × 4, 512, 2, 1	16 × 16 × 512
	Layer 2	BatchNorm	16 × 16 × 512
	Layer 3	ReLU	16 × 16 × 512
	Layer 4	Conv 2D 3 × 3, 512, 1, 1	16 × 16 × 512
	Layer 5	BatchNorm	16 × 16 × 512
	Layer 6	ReLU	16 × 16 × 512
	-	output layer 1 + output layer 6	16 × 16 × 512
	Layer 7	ConvTr 2D, 4 × 4, 256, 2, 1	32 × 32 × 256
	Layer 8	BatchNorm	32 × 32 × 256
	Layer 9	ReLU	32 × 32 × 256
	Layer 10	Conv 2D 3 × 3, 256, 1, 1	32 × 32 × 256
	Layer 11	BatchNorm	32 × 32 × 256
	Layer 12	ReLU	32 × 32 × 256
	-	output layer 7 + output layer 12	32 × 32 × 256
	Layer 13	ConvTr 2D, 4 × 4, 128, 2, 1	64 × 64 × 128
	Layer 14	BatchNorm	64 × 64 × 128
	Layer 15	ReLU	64 × 64 × 128
	Layer 16	Conv 2D 3 × 3, 128, 1, 1	64 × 64 × 128
	Layer 17	BatchNorm	64 × 64 × 128
	Layer 18	ReLU	64 × 64 × 128
	-	output layer 13 + output layer 18	64 × 64 × 128
	Layer 19	ConvTr 2D, 4 × 4, 64, 2, 1	128 × 128 × 64
	Layer 20	BatchNorm	128 × 128 × 64
	Layer 21	ReLU	128 × 128 × 64
	Layer 22	Conv 2D 3 × 3, 64, 1, 1	128 × 128 × 64
	Layer 23	BatchNorm	128 × 128 × 64
	Layer 24	ReLU	128 × 128 × 64
	-	output layer 19 + output layer 24	128 × 128 × 64
	Layer 25	ConvTr 2D, 4 × 4, 3, 2, 1	256 × 256 × 3
	Layer 26	Sigmoid	256 × 256 × 3
Discriminator	Layer 1	Conv2D, 4 × 4, 64, 2, 1	128 × 128 × 64
	Layer 2	Leaky ReLU (0.2)	128 × 128 × 64
	Layer 3	Conv2D, 4 × 4, 128, 2, 1	64 × 64 × 128
	Layer 4	BatchNorm	64 × 64 × 128
	Layer 5	Leaky ReLU (0.2)	64 × 64 × 128
	Layer 6	Conv2D, 4 × 4, 256, 2, 1	32 × 32 × 256
	Layer 7	BatchNorm	32 × 32 × 256
	Layer 8	Leaky ReLU (0.2)	32 × 32 × 256
	Layer 9	Conv2D, 4 × 4, 512, 2, 1	16 × 16 × 512
	Layer 10	BatchNorm	16 × 16 × 512
	Layer 11	Leaky ReLU (0.2)	16 × 16 × 512
	Layer 12	Conv2D, 4 × 4, 512, 2, 1	8 × 8 × 512
	Layer 13	Sigmoid	8 × 8 × 512

Table 10: Architectural details of CAVGA-D_u and CAVGA-D_w. The notation in each row is as follows: operation, filter h $\times$ filter w, number of filters, stride, pad. W.S. denotes the additional layers for the weakly supervised setting. ConvTr 2D denotes transpose convolution layer, Conv 2D denotes convolution layer

network	layer name	layer dimensions	output dimensions
Encoder	Layer 1	Conv2D, 4 × 4, 64, 2, 1	128 × 128 × 64
	Layer 2	Leaky ReLU (0.2)	128 × 128 × 64
	Layer 3	Conv2D, 4 × 4, 128, 2, 1	64 × 64 × 128
	Layer 4	BatchNorm	64 × 64 × 128
	Layer 5	Leaky ReLU (0.2)	64 × 64 × 128
	Layer 6	Conv2D, 4 × 4, 256, 2, 1	32 × 32 × 256
	Layer 7	BatchNorm	32 × 32 × 256
	Layer 8	Leaky ReLU (0.2)	32 × 32 × 256
	Layer 9	Conv2D, 4 × 4, 512, 2, 1	16 × 16 × 512
	Layer 10	BatchNorm	16 × 16 × 512
	Layer 11	Leaky ReLU (0.2)	16 × 16 × 512
	Layer 12	Conv2D, 4 × 4, 512, 2, 1	8 × 8 × 512
	Layer 13	BatchNorm	8 × 8 × 512
	Layer 14	ReLU	8 × 8 × 512
	W.S.: Layer 15	Flatten	32768
	W.S.: Layer 16	Linear	2
	W.S.: Layer 17	Softmax	2
Decoder	Layer 1	ConvTr 2D, 4 × 4, 512, 2, 1	16 × 16 × 512
	Layer 2	BatchNorm	16 × 16 × 512
	Layer 3	ReLU	16 × 16 × 512
	Layer 4	ConvTr 2D, 4 × 4, 512, 2, 1	32 × 32 × 512
	Layer 5	BatchNorm	32 × 32 × 512
	Layer 6	ReLU	32 × 32 × 512
	Layer 7	ConvTr 2D, 4 × 4, 256, 2, 1	64 × 64 × 256
	Layer 8	BatchNorm	64 × 64 × 256
	Layer 9	ReLU	64 × 64 × 256
	Layer 10	ConvTr 2D, 4 × 4, 128, 2, 1	128 × 128 × 128
	Layer 11	BatchNorm	128 × 128 × 128
	Layer 12	ReLU	128 × 128 × 128
	Layer 13	ConvTr 2D, 4 × 4, 64, 2, 1	256 × 256 × 64
	Layer 14	BatchNorm	256 × 256 × 64
	Layer 15	ReLU	256 × 256 × 64
	Layer 16	ConvTr 2D, 1 × 1, 3, 1, 0	256 × 256 × 3
	Layer 17	Sigmoid	256 × 256 × 3
Discriminator	Layer 1	Conv2D, 4 × 4, 64, 2, 1	128 × 128 × 64
	Layer 2	Leaky ReLU (0.2)	128 × 128 × 64
	Layer 3	Conv2D, 4 × 4, 128, 2, 1	64 × 64 × 128
	Layer 4	BatchNorm	64 × 64 × 128
	Layer 5	Leaky ReLU (0.2)	64 × 64 × 128
	Layer 6	Conv2D, 4 × 4, 256, 2, 1	32 × 32 × 256
	Layer 7	BatchNorm	32 × 32 × 256
	Layer 8	Leaky ReLU (0.2)	32 × 32 × 256
	Layer 9	Conv2D, 4 × 4, 512, 2, 1	16 × 16 × 512
	Layer 10	BatchNorm	16 × 16 × 512
	Layer 11	Leaky ReLU (0.2)	16 × 16 × 512
	Layer 12	Conv2D, 4 × 4, 512, 2, 1	8 × 8 × 512
	Layer 13	Sigmoid	8 × 8 × 512

5 Qualitative results on MVTAD, mSTC and LAG datasets

We compare CAVGA-R_u and CAVGA-R_w with the baseline methods on MVTAD and mSTC datasets and present additional anomaly localization results on the MVTAD, mSTC and LAG datasets. Fig. 1 illustrates the ability of CAVGA-D_u to localize the anomalous regions on the LAG dataset. From Fig. 2, Fig. 3, Fig. 4 and Fig. 5, we see that CAVGA-R_u and CAVGA-R_w outperform the baselines in localizing the anomaly and that CAVGA-R_w has better localization performance than CAVGA-R_u on the MVTAD dataset. From Fig. 6 and Fig. 7 we see that CAVGA-R_u and CAVGA-R_w have better anomaly localization as compared to the baseline methods and that CAVGA-R_w surpasses CAVGA-R_u in its ability to localize the anomaly on the mSTC dataset.

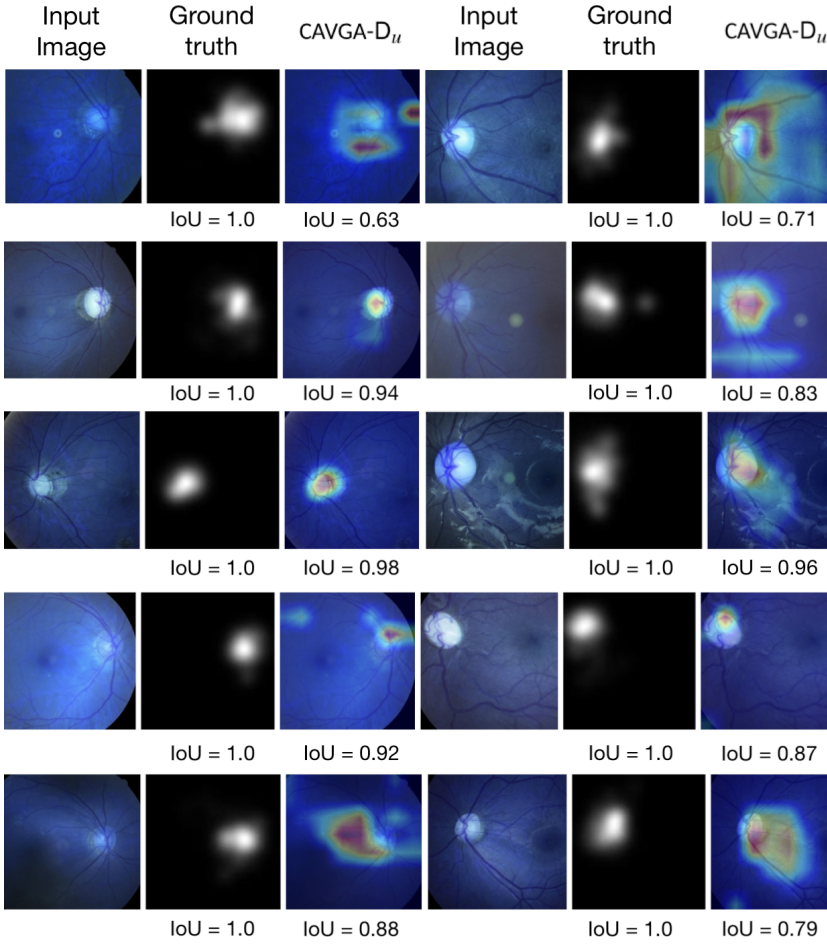

Fig. 1: Qualitative results for anomaly localization of CAVGA-D$_u$ on the LAG dataset. The anomalous attention map depicts the localization of the anomaly in the image.

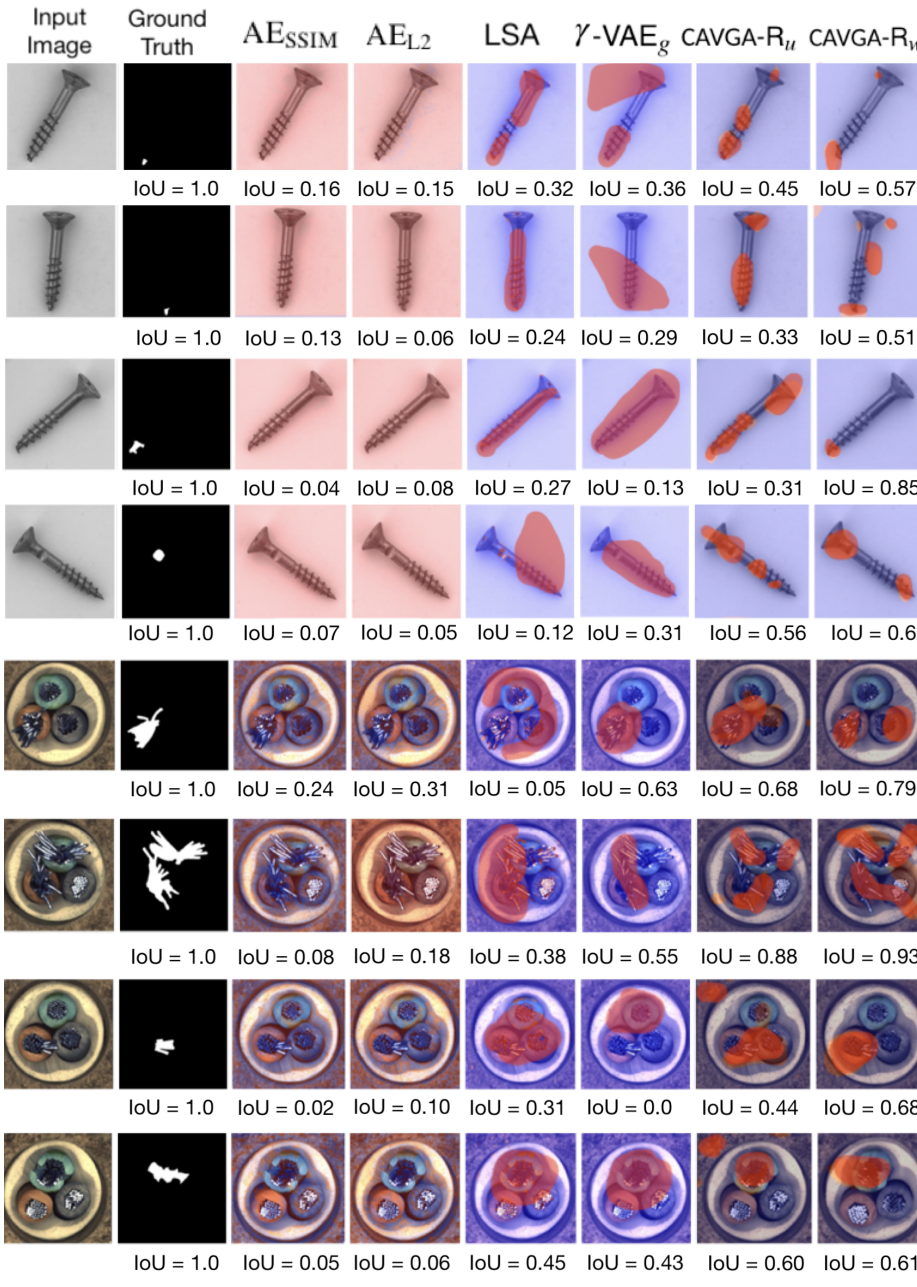

Fig. 2: Qualitative comparison of anomaly localization of CAVGA-R$_u$ and CAVGA-R$_w$ with baseline methods on the MVTAD dataset. The anomalous attention map (in red) depicts the localization of the anomaly in the image.

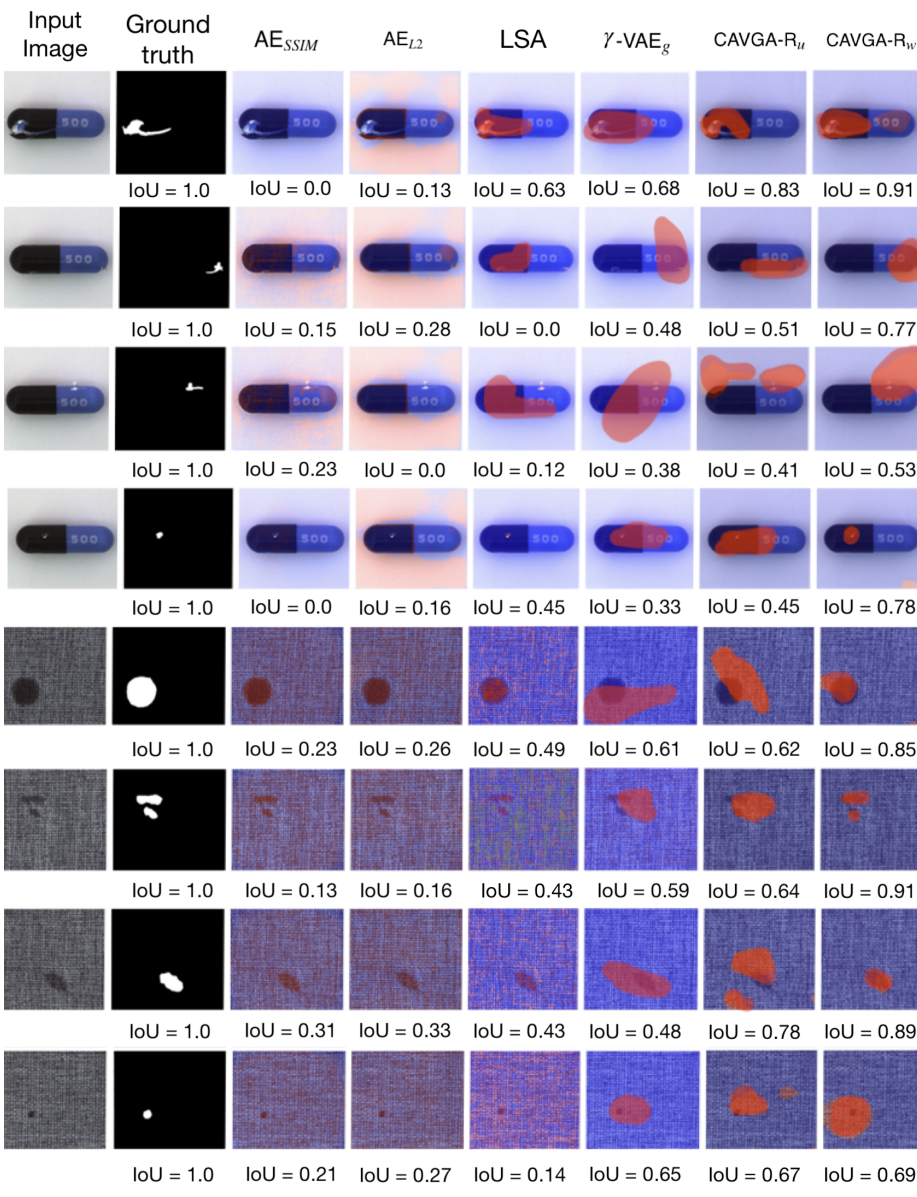

Fig. 3: Qualitative comparison of anomaly localization of CAVGA-R$_u$ and CAVGA-R$_w$ with baseline methods on the MVTAD dataset. The anomalous attention map (in red) depicts the localization of the anomaly in the image.

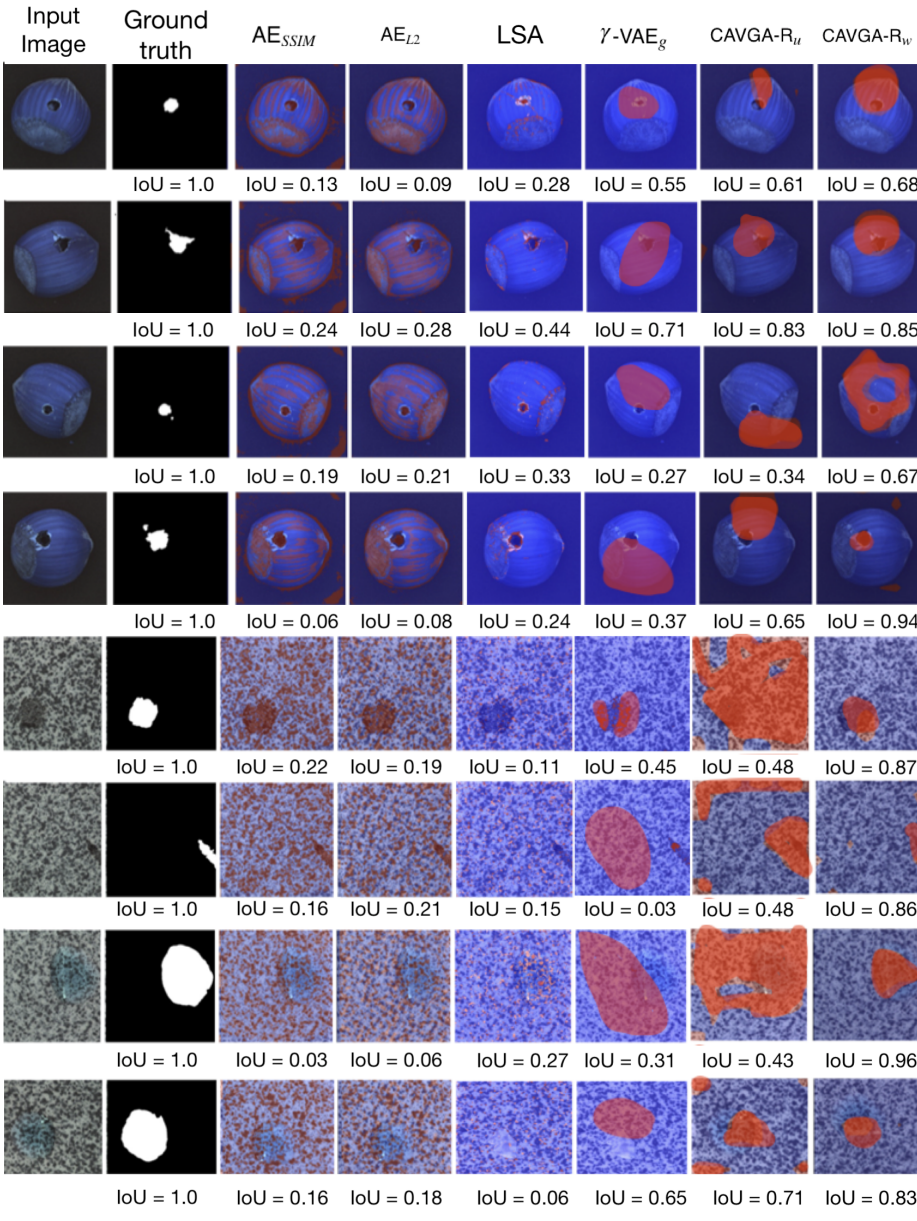

Fig. 4: Qualitative comparison of anomaly localization of CAVGA-R$_u$ and CAVGA-R$_w$ with baseline methods on the MVTAD dataset. The anomalous attention map (in red) depicts the localization of the anomaly in the image.

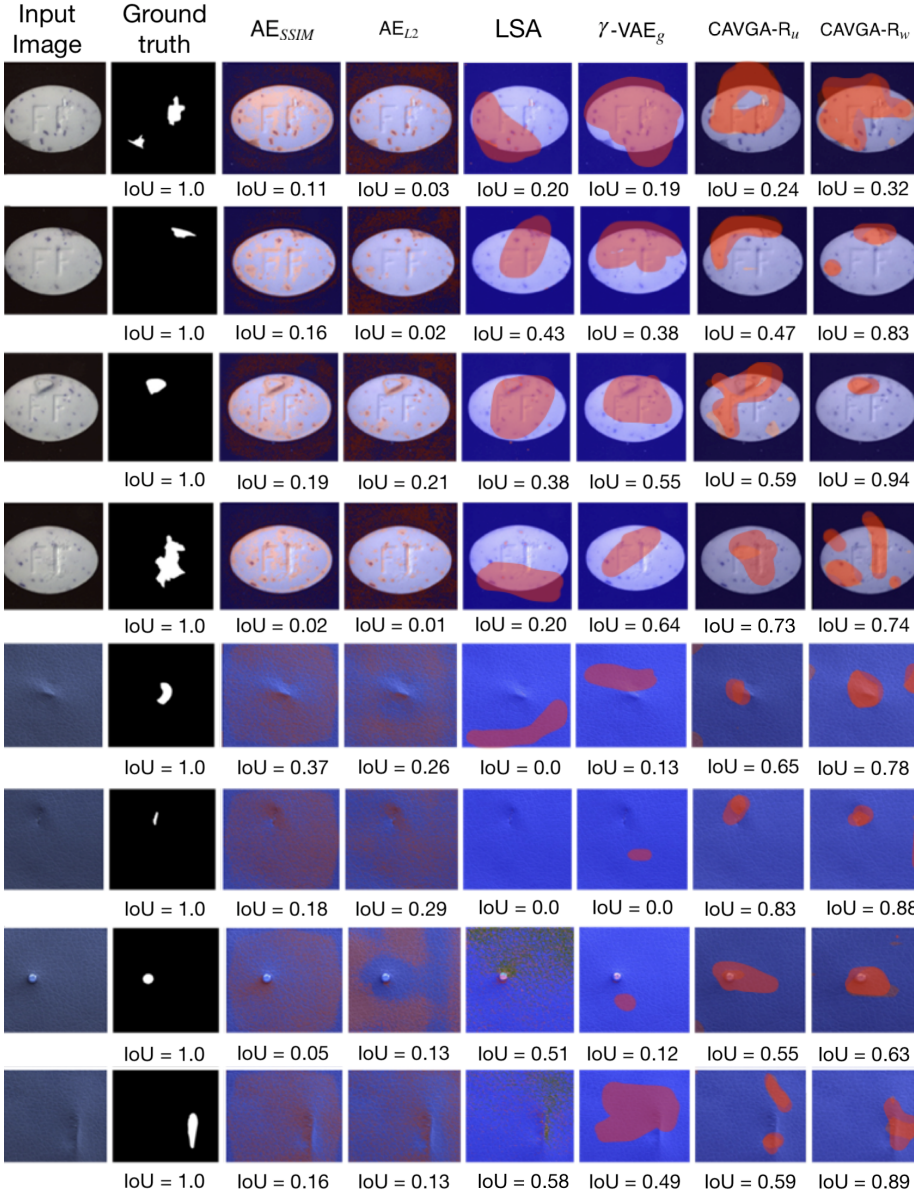

Fig. 5: Qualitative comparison of anomaly localization of CAVGA-R$_u$ and CAVGA-R$_w$ with baseline methods on the MVTAD dataset. The anomalous attention map (in red) depicts the localization of the anomaly in the image.

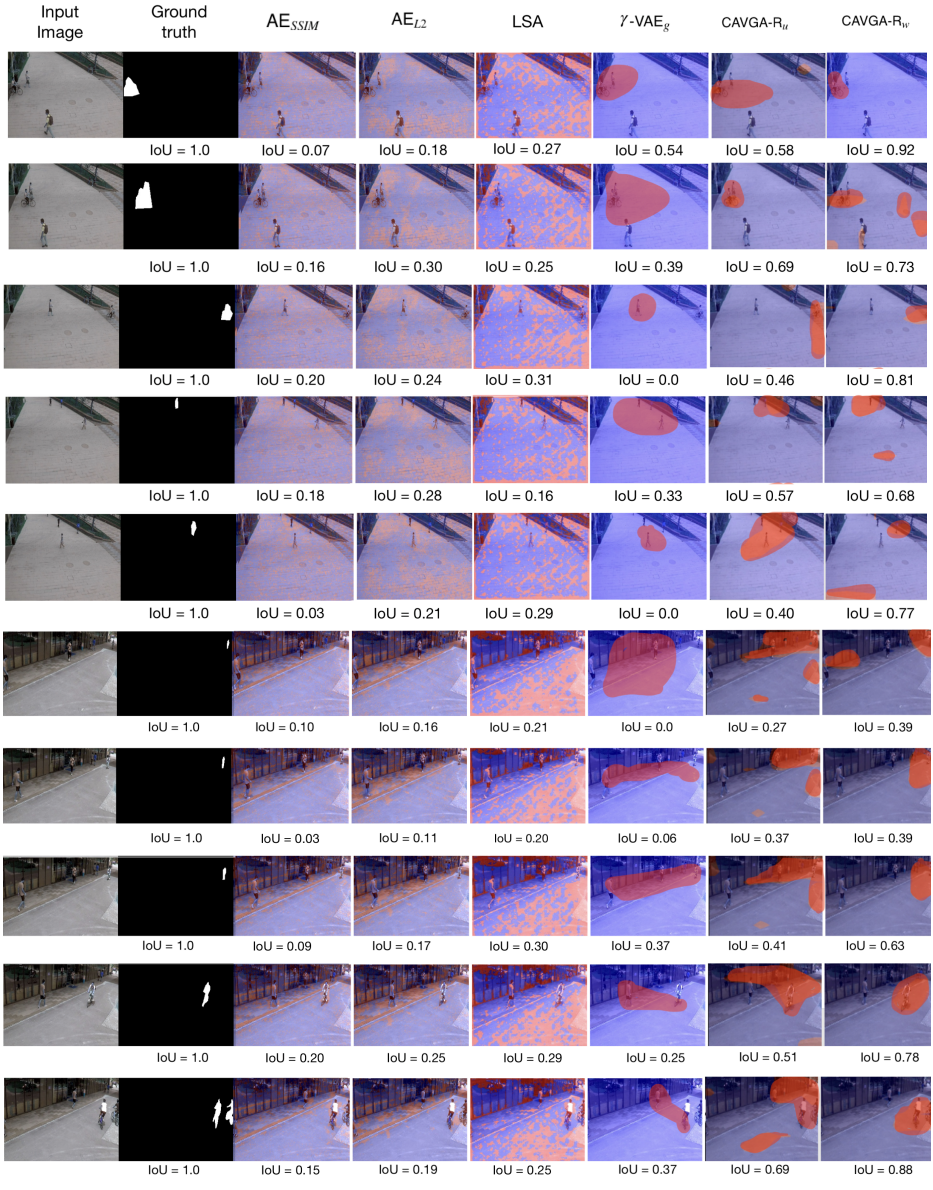

Fig. 6: Qualitative comparison of anomaly localization of CAVGA-R$_u$ and CAVGA-R$_w$ with baseline methods on the mSTC dataset. The anomalous attention map (in red) depicts the localization of the anomaly in the image.

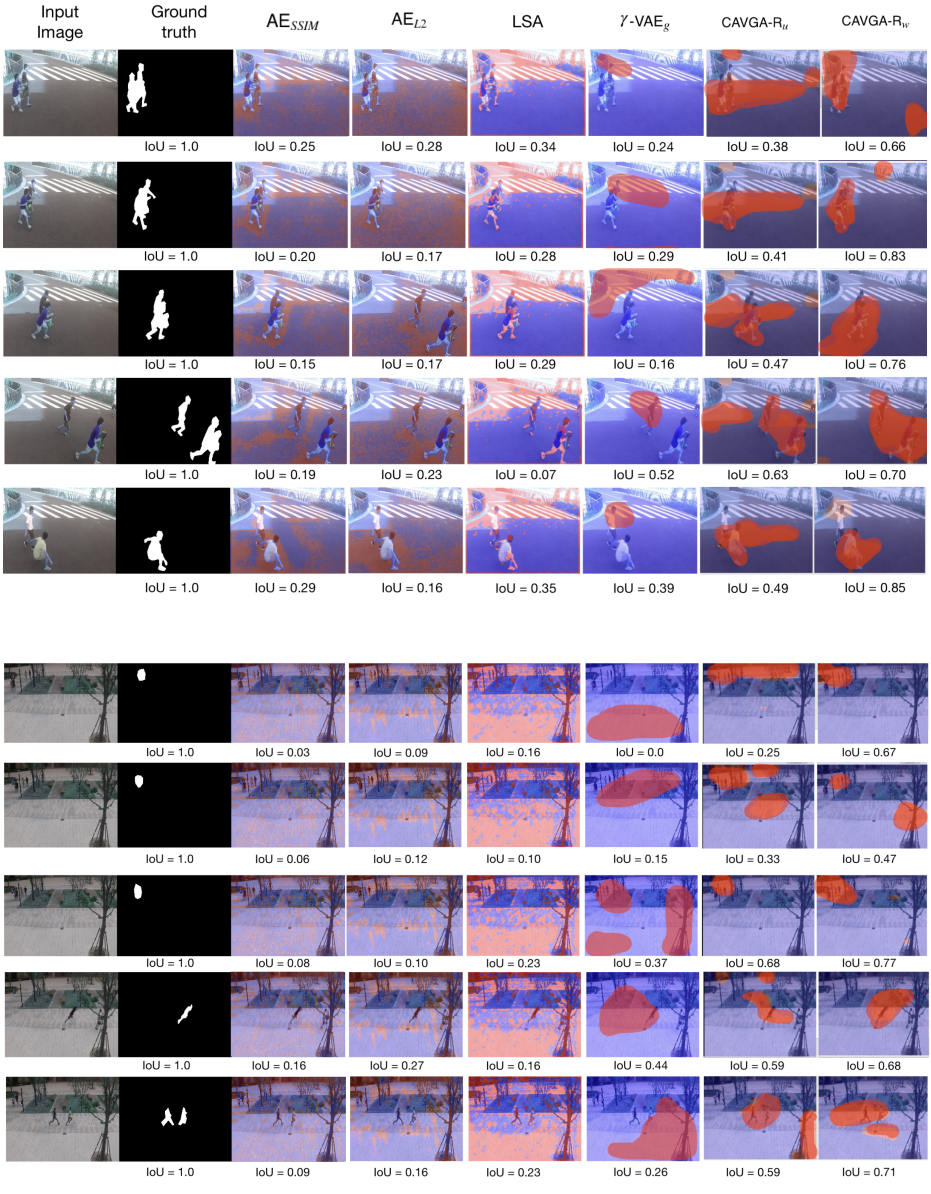

Fig. 7: Qualitative comparison of anomaly localization of CAVGA-R$_u$ and CAVGA-R$_w$ with baseline methods on the mSTC dataset. The anomalous attention map (in red) depicts the localization of the anomaly in the image.

Bibliography

[1] Abati, D., Porrello, A., Calderara, S., Cucchiara, R.: Latent space autoregression for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 481–490 (2019)

[2] Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In: International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP). vol. 5 (2019)

[3] Böttger, T., Ulrich, M.: Real-time texture error detection on textured surfaces with compressed sensing. Pattern Recognition and Image Analysis **26**(1), 88–94 (2016)

[4] Dehaene, D., Frigo, O., Combrexelle, S., Eline, P.: Iterative energy-based projection on a normal data manifold for anomaly localization. International Conference on Learning Representations (2020)

[5] Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Hengel, A.v.d.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1705–1714 (2019)

[6] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-VAE: Learning basic visual concepts with a constrained variational framework. International Conference on Learning Representations **2**(5), 6 (2017)

[7] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)

[8] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[9] Li, X., Kiringa, I., Yeap, T., Zhu, X., Li, Y.: Exploring deep anomaly detection methods based on capsule net. International Conference on Machine Learning 2019 Workshop on Uncertainty and Robustness in Deep Learning (2019)

[10] Liu, W., Li, R., Zheng, M., Karanam, S., Wu, Z., Bhanu, B., Radke, R.J., Camps, O.: Towards visually explaining variational autoencoders. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)

[11] Napoletano, P., Piccoli, F., Schettini, R.: Anomaly detection in nanofibrous materials by CNN-based self-similarity. Sensors **18**(1), 209 (2018)

[12] Sabokrou, M., Pourreza, M., Fayyaz, M., Entezari, R., Fathy, M., Gall, J., Adeli, E.: Avid: Adversarial visual irregularity detection. In: Asian Conference on Computer Vision. pp. 488–505. Springer (2018)

[13] Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging. pp. 146–157. Springer (2017)

[14] Steger, C.: Similarity measures for occlusion, clutter, and illumination invariant object recognition. In: Joint Pattern Recognition Symposium. pp. 148–154. Springer (2001)

[15] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)